論文

教育改善とカリキュラム実質化のための単位認定試験の分析手法

An Analysis Method of Mastery Test for Faculty Development and Realizing Computer Science Curricula

伊藤 克亘　（法政大学情報科学部教授）
荒川 傑　（株式会社ノーチラス・テクノロジーズ）
佐々木 晃　（法政大学情報科学部准教授）
廣津 登志夫（法政大学情報科学部教授）

キーワード
単位の実質化、評価、単位認定試験、分析、教育改善

要旨
効果的かつ継続的な教育改善には、客観的な指標の活用が不可欠である。情報科学部では、授業の達成度を担保するために単位認定試験（MT）を導入した。本稿では、教育改善の指針となる指標を得るための、MTの分析方法を提案する。具体的には、統計的な検定方法を用い、MTの問題のばらつき、学生の分布、クラスごとの分布などを検定する。実際の分析結果を用いた教育改善の例も紹介する。

1．はじめに

社会情勢の変化などにより、学部教育においては、学士の質保証が求められている。無論、学士力とは総合的な力であるべきであるが、大学の教育の根幹をなすのは、いうまでもなく授業であり、それぞれの科目で達成したことの積上げが学士力の中心となるべきであろう。

教員側からこの学士力を見た場合は、最も重要となるのは、それぞれの科目で、個々の学生が何を修得できたかではなく、単位を取得した合格者が共通して何を修得できたか、である。

情報科学部では、各科目の達成度を担保するために2011年から定期試験とは別に単位認定試験（MT: MasteryTest）の導入を行い、実施科目を増やしてきている。本稿では、MTの設計や授業を中心とした教育改善のためのMTの分析方法について提案し、実際の分析を紹介する。

2．単位認定試験（MT）の設計

理系の学部において、伝統的に採用されている積上げ式のカリキュラムを効果的に運用するには、それぞれの科目は、後続の科目の実施・運用に支障を来さないために必要な一定の達成度を保証すべきである。

一方、大学の科目では、合格点は60点程度であることが多い。このことは、単独の科目の判定ではそれほど不都合はない。しかし、カリキュラム上後続科目と連携しなければならない基礎科目では問題がある。そのような科目では、

合格した学生がどの項目をどの程度修得しているかを明確にしなければならない。60%という正解率では、中級層以下で項目ごとの正解率が50%を下回る項目が多数生じてしまう。そのような項目は、後続科目で再度取り上げなければならず、「積み上げ」が困難になる。

また、その科目のカリキュラムへの貢献を明確にするには、「何を教えるか」だけでなく、その科目の合格者は「何を修得できたか」を重視すべきである。特に、合格者が共通して修得できたことを保証する項目（MR: Minimum Requirements）が重要である。MRとは、言い換えれば、学生に最低限修得させる項目である。

この要求を満たすため、MRの達成度だけを評価する単位認定試験（MT: Mastery Test）を導入する。従来の定期試験は、「評定試験」とする。認定試験は、後続科目に必要な最低限の項目を当該科目で達成したことを保証する。一方で、評定試験は、単位を認定された学生が「A+」「A」から「C」の間のどのような能力があるかを判定するものである。授業形態によっては、クラスによって評定試験は違ってもよい。

また、学生の能力や学習活動およびその成果には、試験では測りにくいものもありえる。したがって、最終的な評定は、授業中の所見やいわゆる平常点を考慮しても構わない。

これらの前提で、MTを次のように設計した。

- MRを全て含む
- 合格ラインは90%程度とし、合格しないと評定試験は受験させない（MTを何度も受験させる）。このことは、学生にも告知する。ただし、何回受験できるかは明らかにしない
- 不合格だった場合に、未達成個所を自覚し単位取得のために復習する余地を与える
- 同じ授業名の科目では同一のMTを実施する

この方針に従ってMTを作成するためには、まず、MRを明確にしなければならない。そのために、教材から単元ごとに達成目標を定める。この達成目標の粒度は、90分の授業あたり５項目程度を目安に、達成目標ごとに独立した例題、練習問題、演習問題が作成できる程度の内容になっているかを判断する。また、達成目標は、「～できるようになる」という内容にする。

この達成目標のリストから、最低限達成すべき項目（MR）の一覧を策定する。個数に制限はないが、実施学期中に複数回受験を許すためには、10週目くらいまでにMRが出そろっている必要がある。このMRを基に、次の手順でMTを設計した。

- MRをグループ化し、出題に割り当てる。しかし、試験時間の制限、出題間で互いにヒントにならないようにする、相関の高い項目の場合は前提となるMRを省略する、など運用上の制約から、取捨選択せざるをえない場合がある。
- MRの達成水準を検討し、出題水準を策定する。MRを達成した学生だけを合格させたいので、合格者層は正解し、不合格者層は不正解となるようなレベルを設定する。難易度は、サンプル問題を事前にレビューすることで水準意識を共有する。
- MTでカバーできなかったMRを明確にする。試験時間や難易度など、運用上の都合から、MTで全てのMRがカバーできるとは限らない。これらの点は、関係者に申し送ることで対処する。例えば、授業担当者であれば、MT 以外の手段で評価することを考慮し、カリキュラム作成者であれば、後続科目での評価を考慮する。また、次年度以降にMTの１回目合格率が向上するなどの改善が見られた場合には、学生から見ると容易に達成できた単元・項目が生じたこととなる。その場合、そのような項目をMTから外すことにより、出題範囲に余裕ができることになる。そのような場合、それまでMTでカバーできなかった項目をMTでカバーすることを考慮することも可能になる。

以上のように、MTは期中に行う試験ではあるが、総合的な実力を問うものである。また、単位取得に不可欠であることは、教員側、学生

側で共有される。したがって、部分的な達成度しか問えないにも関わらず、科目全体の単位認定や評点に用いられる通常の中間試験とは性質が大きく異なる。また、授業ごとに行う小テストのような単解答式の問題を大量に寄せ集めるものとも異なる。

MTは設計理念上、MTを受験する段階の、すなわち、十分に学修した学生はランダムに一定の高い確率で正解することを想定している。仮にその正解確率が0.9だとすると、10題の出題であれば、３回の実施後の不合格者は1.8%となる。15題の出題であれば、13題の正解(87%)で、３回の実施で不合格者が0.6%となる。これらより、約15題出題の2011、2012年は85%を合格ラインとし、10題出題の2013年では90%とした。もちろん、実際の学生は受講者全員が十分に学修するわけではない可能性は高いが、設計時はその実態に関する客観的な指標がなかったためこのようなモデルに基づいて設計した。

さらに、2013年度からは、一部の科目において、運用可能性を拡張するために、自動採点のためにオンライン化した[1)]。

オンライン化は以下のような方針で行なっている。

- 昨年まで実施した筆記試験と同等の合格率
- 授業支援システム（sakai）で自動採点するために選択式とする
- 複数回の実施を見据えたランダム出題
- ランダム出題時にも、内容、難易度を制御できるよう内容と難易度を揃えた問題プールを作成
- 単一回答法の多岐選択形式

また、記述式の試験よりも正答率が高くなってしまう事態を避ける設計方針をとった[1)]。

１年次のプログラミングの入門科目「プログラミング入門１」における問題例を示す。この問題は、授業２回分の単元「配列」「繰返」の以下に示すMRに対応している。

- 配列変数と配列参照を利用したプログラムを書ける
- 配列を引数に取るメソッドを宣言し、それを起動するプログラムを書ける
- 配列を返すメソッドを宣言し、それを起動するプログラムを書ける
- ３種の制御構造について説明できる
- 任意の回数だけ同じ処理を繰り返すプログラムを書ける
- 繰り返し変数を利用するプログラムを書ける
- 配列要素を反復し、集計するプログラムを書ける

メソッド「printAverage」は引数に渡された配列の各要素の値の平均をメッセージダイアログに表示する。
空欄（a）（b）にプログラムの断片を埋めよ。

```
void printAverage(int[] elements) {
  int total=0;
  [(a)]
  JOptionPane.showMessageDialog"
  (null, "平均は"+[(b)]+"です。");
}
```

この問題は当該単元に関する小問４問から構成される大問のうちの１問である。正解は、(a)については次のようになる。

```
for(int i=0;i<elements.length;i++){
  total=total+elements[i];
}
```

(b)については次のようになる。

```
total/elements.length
```

同じMRに対するオンライン試験の問題例は次の通りである。

以下のメソッドfの引数にある配列を指定して起動したところ､戻り値は「3」であった。

```
int f(int[]array){
  int x=0;
  for(int i=0;i<array.length;i++){
    if(array[i]>1){
      x=x+array[i];
    }
  }
  return x;
}
```

このプログラムは、受理できる入力が一意に定まるものではない。これに対し、正解の選択肢として「配列の内容として適切なものは選択肢に一つも含まれていない」を用意した。また､「配列の内容として適切なものは選択肢に二つ以上含まれている」という選択肢も用意した。これらが消去法による正解を排除するための選択肢である。不正解の選択肢としては､「順に『-2, 2, 1』」のような間違った内容のものを４つ用意した。他の問題に対する選択肢のタイプとしては、プログラムの断片などがある。

3. MTの分析

3.1 分析手法

テストの分析には、点数分布が正規分布に従うことを暗黙に用いた平均点などの素朴な手法から、S－P表、項目応答分析など様々な手法がある。

しかし、MTは、設計方針で述べたような特徴があるため、従来の手法で分析するのは不適切である。そこで、次のような分析を行う。

1．まず、問題ごとの難易度に差がないかを検定する。難易度に差があると、結局、難易度の高い問題に対応したMRの達成度が低くなりMRとして機能しなくなる。

2．難易度が比較的適切な問題の和の分布を二項分布の和であてはめる。これにより、学生の達成レベルについて分析できる。

3．難易度が比較的適切な問題の和の分布をクラスごとに差がないか検定する。

MTは全員が合格するまで複数回実施（問題は実施ごとに異なる）しているが、受験人数の関係から、１回目の分析を中心とする。

分析手法の詳細について述べる。

3.1.1 問題ごとの難易度分析

MTの定義からすると、理想的には、MTの問題は以下のような条件を満たすべきである。

- 各問が独立である。
- 各問の難易度は等しい。
- 各問の配点は等しい。
- 各問は正解／不正解で採点されるべきである。

これらに従うと、例えば、一般にテストで多用される部分点や複数の小問から構成される大問などは採用できない。

しかし、この制約を満たすためには、これらの制約を前提とした問題作成の経験や、実施した結果のフィードバックが欠かせない。したがって、運用上は、部分点や小問の採用も妨げない。

そのような状況であっても、理想的なMTを実現するには、理想的な状況が実現されたかどうかを確認する分析手法が必要である。

そこで、問題ごとの難易度分析では、次の前提をおく。

- それぞれの問題の正解率はベルヌーイ分布に従う。

運用上、部分点を認める場合には、分析のときには完答だけを正解として正解率を計算する。

この前提に立つと、それぞれの問題のベルヌーイ分布の成功確率（問題の正解率）θは、実際の試験の受験者数をn、正解者数をxとすると、$\hat{\theta}=x/n$で推定される。ここで難易度の比較をするために、この確率θの信頼係数$1-\alpha$の信頼区間

$$\Pr(\theta_L<\theta<\theta_U) \geq 1-\alpha$$

表1　プログラミング入門1　2013年 MTの問題別正解状況

問題番号	1	2	3	4	5	6	7	8	9	10
正解数	159	174	129	161	132	170	153	166	140	150
正解率	0.91	0.99	0.73	0.92	0.75	0.97	0.87	0.95	0.80	0.86

となる区間（θ_L, θ_U）を問題ごとに推定する。この区間は、二項確率の計算に基づく正確法によって次のように求められる[2)]。

$$\theta_L = \frac{x}{x+(n-x+1)F_{2(n-x+1),\,2x}(\alpha/2)}$$

$$\theta_U = \frac{x}{x+1+(n-x)F_{2(x+1),\,2(n-x)}(\alpha/2)}$$

ここで、$F_{v1,v2}(\alpha/2)$ は、自由度（$v1$,$v2$）のF分布の上側$100\alpha/2\%$点を表す。

問題ごとに区間推定し、中間の正解率（θ）を持つ問題の信頼区間と重なっていない信頼区間を持つ問題を難易度が異なる問題とみなす。

なお、大問形式の出題方法をとった場合は、得点分布はヒストグラムとなる。ただし、受験者の人数がそれほど多くなく、得点の種類もそれほど多くなく（離散的で）、MTの性質上、満点が多い分布となるため、正規分布を仮定するのは不適切である。したがって､クラスカル･ウォリス検定に基づく多重比較で、上記のような難易度の分析を行う。

3.1.2　得点分布の二項分布の和によるあてはめ

MT設計時の想定通りであれば、学生はそれぞれの問題に対し$\theta=0.9$程度の正解確率でランダムに正解する。その場合、受験者全体の得点分布は二項分布B（x, θ）に従う。

しかし、実際には、学生の学修状況や準備状況は均一ではなく、大別してもいくつかの層に分けられる。そこで、それらの層について分析するため、得点分布にあてはまる二項分布の和による分布を推定する。推定には、分布数、初期値を適当にあたえて最尤推定する。

この際、MTとして適切に設計されていない問題は除外する。

3.1.3　クラスごとの得点分布の比較

前述のようにMTの得点分布は、正規分布を仮定するのは不適切である。そこで、２クラスの場合は、コルモゴロフ－スミルノフ検定、多クラスの場合は、クラスカル・ウォリス検定に基づいた多重比較を行う。

この場合、帰無仮説は「各群で差がない」となり、対立仮説は「各群で差がある」となる。MTの分析においては、便宜上、帰無仮説が棄却されない場合は、差がないとみなすことにする。

仮説検定の有意水準は記さない限り0.05とする。

3.2　分析例

分析例として、「プログラミング入門１」の2013年のMTの分析結果を示す。このMTは、10問から構成されている。受験者総数は175名である（表１）。

問題の難易度を分析した結果、図１のような結果が得られた。

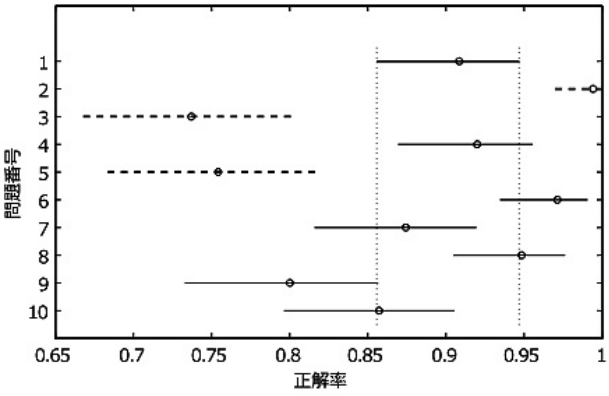

図1　プログラミング入門1
2013年 MTの問題の難易度の分析

丸印がそれぞれの問題の正解率である。５番目の正解率の問題（問題１）を基準として、信

頼区間が重ならない問題を破線で示している。

問題２が簡単すぎ、問題３，５は難しすぎることがわかる。これらの問題を除外した場合の平均正解率は0.90であり、ほぼ想定通りになっていた。問題２は、問題形式がストレートであり、１つの条件だけで選択肢を選ぶことが可能なものであった。一方、問題３，５は、頭の中だけで考えるには複雑な問題であることが特徴であった。実際に、計算用紙などに実行中のイメージを書いたりすれば、それほど複雑ではないが、地道に動作を追ったり、抽象的に理解して頭の中だけで処理することができていないと推測される。これらの知見は、次回以降のMTの設計や授業構成・授業方法の改善に活かす。

次に、学生の得点の分布を二項分布の和であてはめる。問題2, 3, 5を除いた場合（以後、「修正済」と参照する）には、$0.77B(7, 0.96)+0.23B(7, 0.69)$となった。ヒストグラム（黒）と推定した分布に基づくヒストグラム（白）を並べたものを図２に示す。

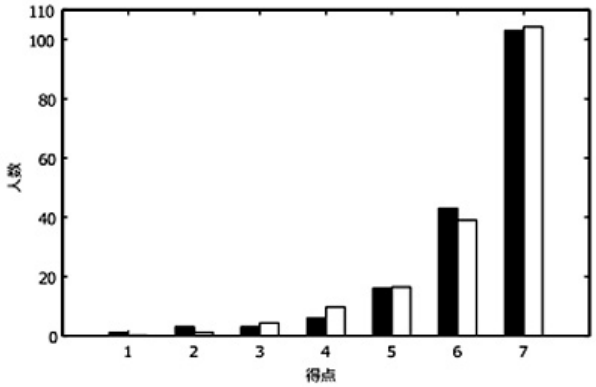

図2　プログラミング入門1
2013年 MTの二項分布によるあてはめ
（修正済）

このあてはめの結果から、学生の77%程度は、想定よりもかなり高い正解率（0.96）を示していることがわかる。一方で、残りの学生は、試験に対して十分な準備ができていないと推測される。準備が十分でない学生の比率は、授業を行っている教員の印象にも比較的合致している。また、十分に準備できていない学生の中には、授業で十分に理解できていない学生も混ざっているため、課外補習を行うことでMTを4回実施する中で全員合格させることができた。

また、このMTは簡単すぎるため、MRの追加やMTの問題の見直しが必要であることが示唆される。

学生の修正済の得点分布のクラス別（４クラス）の多重比較検定の結果、クラスごとの分布に違いがあるという仮説は棄却された。したがって、意図通りに設計されたMTの範囲では、教員による達成度に差がない可能性が示唆される。修正済のクラスごとの得点のヒストグラムを図３に示す。

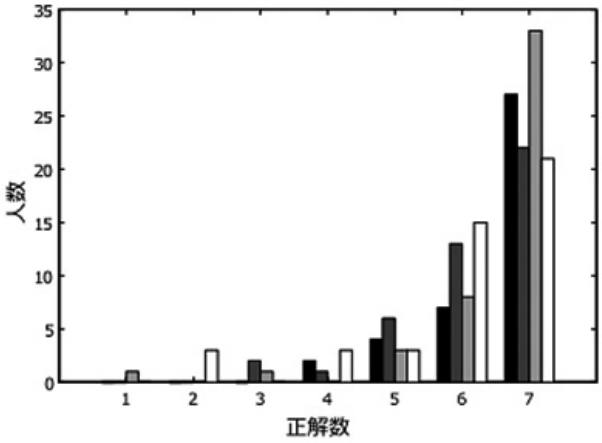

図3　プログラミング入門1
2013年 MTのクラスごとのヒストグラム
（修正済）

ただし、ここまでに紹介したプログラミング入門1は、2011年度からMTを実施しており、３年目であるため、全体としてもかなり想定通りの設計になっている。難易度が異なる問題を除外しない場合でも、平均の正解率は0.88である。二項分布によるあてはめの結果は図４のようになる（$0.78B(10, 0.93)+0.22B(10, 0.67)$）。

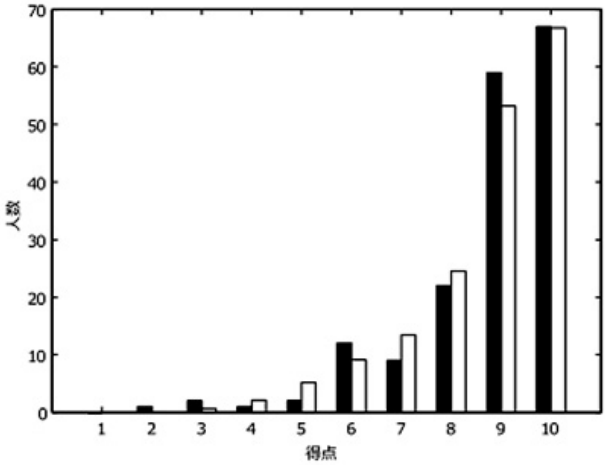

図4　プログラミング入門1
2013年 MTの二項分布によるあてはめ（全体）

修正済の場合よりも若干あてはまりが悪いが、学生層などの分析結果はほぼ同じである。

3.3　考察

提案した分析手法で、授業改善、教育改善につながる以下のような項目についての知見が得られることが期待できる。

- MTの設計の問題
- 授業設計の問題
  - －MRの適切性
- 学生のレベル
- 授業の問題
  - －教員ごとのばらつき
  - －授業方法の問題
  - －授業構成の問題

これらの知見が得られることで、例えば、期中であれば、期中の補習の実施に活かせる。また、長期的には、授業改善、カリキュラム改善などに活用できる。

実際のMTの分析例を用いて説明する。

上記の入門科目の後続科目である、２年生向けのプログラミング科目のMTでは、まず、問題の難易度分析で２問が高難易度として除外された。高難易度の問題の一つは、「妥当な解がない」という選択肢が正解だったものである。当て推量が効かず、全選択肢を吟味しないといけないため正解率が低くなった。もう一つは、MR自体の難易度が高く、さらに、MTの直前に教えられたものである。このような場合、習熟させるために授業構成を変更するという対応も考えられるが、この項目はMRの依存関係で授業構成を工夫することも困難となっている。このような項目でも、MTで正解できなかったことにより学習の必要性を自覚させるという教育効果がある。

これらの問題を除いて残りの問題の合計点を二項分布の和であてはめた結果$0.5B(8, 0.83)+0.25B(8, 0.74)+0.25B(8, 0.47)$となった。

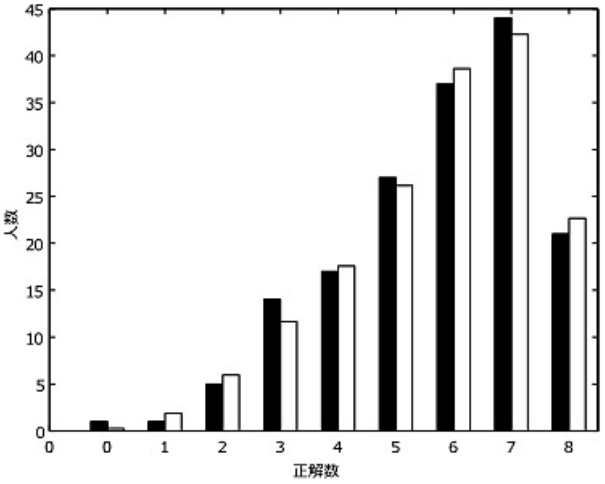

図5　プログラミング科目MTの二項分布によるあてはめ（修正済）

このように、正解率が想定よりも低い場合には、まず、この年度の教育効果を保証することが最優先である。当該科目においては、授業でMTの問題の解説をして、MRの達成の徹底を図った。MT自体の評価としては、まず、問題の適切性が疑われる。しかし、この科目は前章の入門科目と類似性が高く、長年開講してきている科目であるため、問題自体には、それほど問題はないだろう。現状では、１回目の合格率が低い（10%未満）。しかし、その最大の要因は、準備が不十分な学生の割合が多いためであり（半数程度と推定される）、現状の難易度であれば、その層から運よく合格する確率がほとんどないため、MTはそのままで、学生の学修

態度の改善に取り組むべきだと考えている。

準備が不十分な学生の対策例をいくつか紹介する。１年生の科目においては、MTの開始年度でそのような事態が発覚してからは、MT直前の時期での授業中の学生の理解度を観察して、理解度がMT合格に不十分な学生を想定した補習を授業外に行っている。MTを未体験の段階での補習であるので、理解度が高めの学生も念のため出席している。

本稿で問題となっている２年生科目については、MTの難易度や位置付けは理解できているという前提から、準備が不十分であることを自覚させるために明示的な対応は敢えておこなっていない。さらに多くの科目でMTを実施する中で自然と準備できるスキルを自発的に身につけることを期待している。

正解率が想定より低すぎるような事態は、従来の定期試験でも生じることはあっただろう。しかし、そのような場合に、これまでは、合格ラインを下げたり、次年度以降に内容を削減したりする、ということが起きていたのではないだろうか？ 単独の科目では、そのようなことは問題となりにくいが、積み上げ式のカリキュラムでは、そのような恣意的な運用は避けるべきである。したがって、MTの評価およびそれを受けての改善は、カリキュラム運営主体（学部執行部やFD委員など）が行うべきである。

上記の入門科目の2011年度（初めてのMTの実施）のときには、４クラス中、１クラスだけが、平均点が有意に異なる（低い）という検定結果が得られた。その理由を推定するため、様々な分析をおこなった。授業改善アンケートにおいては、出席率（平均0.9）や時間外学習時間（平均2.1時間から２．３時間）は差がなかった。さらに、満足度（この授業を履修してよかったと思いますか。）については、４クラスの中で最も高く0.8であった。しかし、実は、回収率が、４クラスの中で、突出して低かったのである。この事実は、担当教員の「他の年に比べても出席状況が悪かった」との印象にも合致していた。このように授業改善アンケートでは（明示的には）明らかにならないクラスの違いを明らかにした。また、別の視点からは、授業改善アンケートについては、回収率が非常に大きな情報を持つ、もしくは、集計結果を回収率を考慮して正規化するなどの必要があるといえる。

別の専門基礎科目のMTでは、２クラス実施の状況で、クラスごとの違いが有意であるという結果が得られた。問題ごとの正解率の差も大きかった。中でも最も正解率が低かった単元では、特定の部分の正解率でクラス間の差が大きかった。実は、この科目では、片方のクラスの担当教員がMTの問題を作成したのだが、そのクラスの正解率が高かった。これらのことから、当該科目では、MRに対するMTの設計について不十分である可能性、もしくは、担当教員のMRやMTの理解が不十分であった可能性が示唆される。（この科目では、MRは授業担当教員以外の教員と外部で策定した。）上記のプログラミング入門科目では、MTの作成は外部に委託している。MTは外部や担当教員以外の教員が設計に強く関与することで、MRの選定やMTの設計と、クラスごとの授業の実施内容の分離が実現するのではないだろうか。

4. おわりに

本稿では、統計的な検定手法に基づいた単位認定試験（MT）の分析手法を提案した。この分析結果は、MTの設計の確認、授業改善のためのPDCAサイクルのチェックの材料として活用できることを示した。

教育の現場では、ときにテストの有効性に疑問が呈されることがある。しかし、古くから評価に用いられてきたことからもわかるように確実にテストで測れることはある。テストは実施コストが非常に高い面もあるが、テストで測れるものはテストで測ればよい、もしくは、テストで測れるものはテストで測るべきではないだろうか。本稿の分析例で示したように、授業ア

ンケートでは、回収率の低さ（といっても、全学平均から比較して低いわけでない）ゆえか、実態とは正反対の結果にも見えていたクラスの問題点を明らかにする場合もある。テストの利点は、学生が真剣であること、回収率が他のデータよりもはるかに高いことにあるだろう。

もちろん、テストも万能ではないので、テストで測定できない学生の能力については、テスト以外の評価基準や測定方法を導入／開発すべきである。

【謝辞】

一部は、2012年度「特色あるFDへの取組み」で実施された。

【参考文献】

1）佐々木晃，伊藤克亘（2013/11）：「情報系カリキュラムを通じた教育の実質化のためのICTを活用した導入科目の達成度保証」『ICT活用教育方法研究』vol.15，no.1.

2）岩崎学（2010）：『カウントデータの統計解析』朝倉書店.